



SAFA

PRÁCTICA 1 PYCHARM

EDD | José Ignacio Naranjo García

# Práctica 1 PyCharm

---

## Actividad 1

Escribe un programa que almacene las asignaturas de un curso (por ejemplo Informática, Francés, Filosofía, Ética y Álgebra) en una lista y la muestre por pantalla el mensaje “Yo estudio <asignatura>”, donde “asignatura” es cada una de las asignaturas de la lista.

Una vez terminado, documenta con el historial de VCS los diferentes estados por los que ha pasado tu código.

## Actividad 2

El programa debe almacenar los vectores (1, 2, 3) y (-1, 0, 2) en dos listas y muestre por pantalla su producto escalar.

$$\text{Nota: } \vec{u} * \vec{v} = u_1 * v_1 + u_2 * v_2 + \dots + u_n * v_n$$

El siguiente código es cercano a la solución, pero contiene errores. Haciendo uso del debugger, explica cómo has encontrado la solución y cuál es esta.

```
a = (1, 2, 3)
b = (-1, 0, 2)
product = 0
for i in range(len(a+1)):
    product = a[i-1]*b[i-1]
print("El producto de los vectores" + str(a) + " y " + str(b) + " es " + str(product))
```

## Actividad 3

El programa debe recibir una cadena de caracteres y devolver un diccionario con cada palabra que contiene y el número de veces que aparece.

Otra función que reciba el diccionario generado con la función anterior y devuelva una tupla con la palabra más repetida y su frecuencia.

Por ejemplo, ante el texto: ‘Como quieres que te quiera si el que quiero que me quiera no me quiere como quiero que me quiera’, nuestro programa ha de devolver:

```
{'Como': 1, 'quieres': 1, 'que': 4, 'te': 1, 'quiera': 3, 'si': 1, 'el': 1, 'quiero': 2, 'me': 3, 'no': 1, 'quiere': 1, 'como': 1}
('que', 4)
```

El siguiente código es cercano a la solución, pero contiene errores. Haciendo uso del debugger, explica cómo has encontrado la solución y cuál es esta.

```

#primera función, devuelve cada palabra y su frecuencia
def count_words(text):
    text = text.split() #divide el texto en palabras
    words = {}
    for i in text: #recorremos cada palabra dividida en text
        if i in words:
            words[i] += 1
        else:
            words[i] = 1
    return words

#devuelve la palabra que más aparece
def most_repeated(words):
    max_word = ''
    max_freq = 0
    for word, freq in words.items():
        if freq > max_freq:
            max_word = word
            max_freq = freq
    return max_word, max_freq

text = 'Como quieres que te quiera si el que quiero que me quiera no me quiere como quiero que me quiera'
print(count_words(text))
print(most_repeated(count_words(text)))

```

## Actividad 4

Escribir un programa que almacene la cadena de caracteres `12345EDD` como contraseña. En una variable, pregunte al usuario por la contraseña hasta que introduzca la contraseña correcta.

## Actividad 5

El programa debe preguntar al usuario por una frase y una letra, y mostrar por pantalla el número de veces que aparece la letra en la frase.

El siguiente código es cercano a la solución, pero contiene errores. Haciendo uso del debugger, explica cómo has encontrado la solución y cuál es esta.

```

frase = input("Introduce una frase: ")
letra = input("Introduce una letra")
contador = 0
for i in frase:
    if i != letra:
        contador += 1
print("La letra '%s' aparece %2i veces en la frase '%s'." % (letra, contador, frase))

```

### Actividad 6

El programa debe almacenar las matrices en una lista y mostrar por pantalla su producto.

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix} \quad y \quad B = \begin{pmatrix} -1 & 0 \\ 0 & 1 \\ 1 & 1 \end{pmatrix}$$

Nota: Para representar matrices mediante listas usar listas anidadas, representando cada vector fila en una lista.

El siguiente código es cercano a la solución, pero contiene errores. Haciendo uso del debugger, explica cómo has encontrado la solución y cuál es esta.

```
a = ((1, 2, 3),
      (4, 5, 6))
b = ((-1, 0),
      (0, 1),
      (1,1))
result = [[0,0],
           [0,0]]
for i in range(len(a)):
    for j in range(len(b[0])):
        for k in range(len(b)):
            result[i][j] += a[j][j] * b[j][i]
for i in range(len(result)):
    result[i] = tuple(result[i])
result = tuple(result)
for i in range(len(result)):
    print(result[i])
```

### Actividad 7

Escribir un programa que almacene en una lista los siguientes precios, 50, 75, 46, 22, 80, 65, 8, y muestre por pantalla el menor y el mayor de los precios.

Utiliza el debugg para solucionar los errores que te hayan podido surgir y documéntalo.

## Actividad 8

El programa debe convertir un número decimal en binario. Otra función, convertirá un número binario en decimal.

El siguiente código es cercano a la solución, pero contiene errores. Haciendo uso del debugger, explica cómo has encontrado la solución y cuál es esta.

```
def to_decimal(n):
    n = list(n)
    n.reverse()
    decimal = 0
    for i in range(len(n)):
        decimal = int(n[i]) * 1 ** i
    return decimal

def to_binary(n):
    binary = []
    while n > 0:
        binary.append(str(n / 2))
        n //= 2
    return ''.join(binary)

print(to_decimal('10110'))
print(to_binary(22))
print(to_decimal(to_binary(22)))
print(to_binary(to_decimal('10110')))
```